User Defined Software Environments - Singularity Containers for HPC

Michael Fink

University of Innsbruck

Introduction

Since the early 2000s, in connection with adoption of computationally intensive methods by an increasing number of scientists and scientific fields, including those not traditionally associated with High Performance Computing, such as Life Sciences and Humanities, HPC has seen a massive growth in the number, size, and complexity of software products with proliferating and sometimes conflicting prerequisites (known colloquially as *Dependency Hell*). The problem is aggravated by conflicting expectations of stability versus innovation, lack of documentation and domain-specific knowledge, and limited human resources.

User mobility between HPC sites is impaired by the effort required to port, test, and optimize software to a new environment. The rapid development and inhomogeneous deployment of software and library versions across computing facilities raise questions of *reproducibility* of scientific results.

Software containers, which are a lightweight alternative to OS virtualization, try to address these problems by bundling software with all needed prerequisites into an isolated environment. Containers can be executed on arbitrary machines with very few dependencies on locally installed software. Archived containers may be used to verify past results if necessary.

Software Containers for HPC

Docker is a wide-spread software container system backed by many software and OS vendors, but is incompatible with the operational requirements and workflows typically found in HPC environments.

Singularity [1, 2] is an open source runtime system for software containers specifically geared towards use on HPC systems, allowing users to build their own portable software environments from various sources, including Docker Hub. Singularity containers, while bringing along their own copy of operating system utilities, libraries, and software, run fully integrated on HPC clusters with full access to these servers' compute resources, user data, and interprocess and internode communication facilities, with no virtualization overhead and under full control of HPC job management systems.

Singularity at the University of Innsbruck

The HPC team [3] of the University of Innsbruck IT-Services department (ZID) has been supporting Singularity since 2017. In my talk, I will present the architecture and functionality of Singularity, describe typical workflows and pitfalls, and report on experiences with containers on our HPC compute clusters.

References

[1] Kurtzer GM, Sochat V, Bauer MW, Singularity: Scientific containers for mobility of compute. PLoS ONE **12(5)** (2017) e0177459. `https://doi.org/10.1371/journal.pone.0177459`

[2] Singularity Homepage `https://www.sylabs.io/singularity`

[3] University of Innsbruck – ZID (IT Services) – HPC Homepage `https://www.uibk.ac.at/zid/systeme/hpc-systeme`